TestData alerting

- Before you begin
- Choose a scenario for alerting
- Create an alert rule
- Example: Predictable Pulse alert
- Example: CSV Metric Values threshold alert
- Test error and no-data conditions
  - No data
  - Errors
  - Intermittent errors
  - Timeouts
- Limitations
  - Template variables aren't supported in alert queries
  - Only time-series scenarios work with alert conditions
  - Streaming scenarios aren't evaluated by the alerting engine

The TestData data source supports Grafana Alerting. You can use it to prototype alert rules, test threshold conditions, and verify notification pipelines without connecting to an external data source.

Before you begin

- Configure the TestData data source.
- Familiarize yourself with alert rules and how to create a Grafana-managed alert rule.

Choose a scenario for alerting

TestData scenarios that return time-series data work with alert rule conditions. Choose a scenario based on the behavior you want to test.

Scenario	Alerting use case
Predictable Pulse	Deterministic on/off pattern. Produces repeatable alert firing and resolving on a fixed cycle.
Random Walk	Non-deterministic time series. Useful for quick prototyping and load testing alert evaluation.
CSV Metric Values	Controlled, fixed values. Test exact threshold boundaries.
Predictable CSV Wave	Custom repeating waveforms. Test complex threshold patterns with precise control.
Random Walk (with error)	Returns both data and an error. Test how alerts handle partial failures.
Error with source	Returns a plugin or downstream error. Test alert evaluation error handling.
Flaky Query	Returns errors intermittently. Test how alerts handle a data source that fails part of the time.
No Data Points	Returns empty results. Test no-data alert conditions.
Slow Query	Introduces a configurable delay. Test alert evaluation timeouts.

Scenarios that produce logs, traces, or streaming data aren't suitable for threshold-based alert conditions.

Create an alert rule

To create an alert rule using TestData:

1. Navigate to **Alerting** > **Alert rules**.
2. Click **New alert rule**.
3. Enter a name for the rule.
4. Select the **TestData** data source.
5. Choose a scenario from the **Scenario** drop-down (for example, **Predictable Pulse**).
6. Configure the scenario options.
7. Add a **Reduce** expression to aggregate the series (for example, **Last** value).
8. Add a **Threshold** expression to define the condition (for example, **Is above** `0.5`).
9. Set the evaluation interval and pending period.
10. Configure notifications and labels.
11. Click **Save rule**.

For detailed instructions, refer to Create a Grafana-managed alert rule.

Example: Predictable Pulse alert

The Predictable Pulse scenario produces a repeating on/off wave that's ideal for testing alert rules because the firing pattern is deterministic and reproducible.

To create a Predictable Pulse alert:

1. Select the **Predictable Pulse** scenario.
2. Configure the pulse options:

Field	Value	Effect
Step	`60`	One data point every 60 seconds.
On Count	`3`	Three consecutive points at the "on" value.
Off Count	`6`	Six consecutive points at the "off" value.
On Value	`1`	Value during the "on" phase.
Off Value	`0`	Value during the "off" phase.

3. Add a **Reduce** expression with function **Last**.
4. Add a **Threshold** expression: **Is above** `0.5`.

With this configuration, the alert fires for 3 minutes (3 x 60s on-points) and then resolves for 6 minutes (6 x 60s off-points), repeating indefinitely. Adjust the step, on count, and off count to control the timing.

Example: CSV Metric Values threshold alert

The CSV Metric Values scenario lets you define an exact sequence of data points, making it useful for testing precise threshold boundaries.

To create a threshold alert with fixed values:

1. Select the **CSV Metric Values** scenario.
2. Enter a comma-separated list of values in the **String Input** field. For example:

```
10,45,70,95,50,20
```

3. Add a **Reduce** expression with function **Last**.
4. Add a **Threshold** expression: **Is above** `80`.

With this input, the series cycles through the values `10, 45, 70, 95, 50, 20`. The alert fires when the reduced value exceeds `80` (the `95` data point) and resolves when it drops back below the threshold.

This approach is helpful for validating that a threshold condition triggers at the exact boundary you expect.

Test error and no-data conditions

TestData includes scenarios designed for testing how alerts respond to failures. Use these to verify that your alert rules handle edge cases correctly before connecting to production data sources.

No data

Select the **No Data Points** scenario. The backend returns an empty result, which triggers the alert rule's **no data** state. Use this to verify that your rule is configured to alert, keep the last state, or resolve when no data is received. Configure this behavior in the alert rule's **No data and error handling** section.

Errors

Select **Error with source** and choose the error type:

- **Plugin**: Simulates a failure in the plugin itself. The alert enters an error state.
- **Downstream**: Simulates a failure in the data source or network. The alert enters an error state with a downstream classification.

You can also use **Random Walk (with error)**, which returns valid data alongside an error, to test how the alerting engine handles partial failures.

Intermittent errors

Select **Flaky Query** to simulate a data source that fails part of the time. Set the **Error rate** to control how often requests fail, and use **Status code** and **Error source** to shape the error. Use this to verify how an alert rule behaves when evaluations succeed and fail across consecutive

intervals, and how the **No data and error handling** settings respond to intermittent failures.

Timeouts

Select **Slow Query** and set the **String Input** field to a duration longer than your evaluation interval (for example, `30s` for a rule evaluated every `10s`). The default delay is `5s`. Use this to verify timeout behavior and confirm that alerts transition to the expected state when queries take too long.

Limitations

When using TestData with Grafana Alerting, be aware of the following limitations.

Template variables aren't supported in alert queries

Grafana evaluates alert rules on the backend without dashboard context. Template variables like `$varname` aren't resolved during alert evaluation. Use fixed values in the scenario options instead.

Only time-series scenarios work with alert conditions

Alert conditions require numeric time-series data. Scenarios that produce logs (Logs), traces (Trace), graphs (Node Graph, Flame Graph), annotations (Annotations), or tables (Table Static) can't be used as alert conditions.

Streaming scenarios aren't evaluated by the alerting engine

The alerting engine evaluates queries at fixed intervals and doesn't consume streaming data. Avoid using Streaming Client, Grafana Live, or Simulation with streaming enabled for alert rules.

□□□ #2
□ □□□ □□□ 2026-10-31 16:04:19 UTC
□ □□□ □□□ 2026-10-31 16:04:21 UTC